

(43)Date of publication of application : 07.09.2001

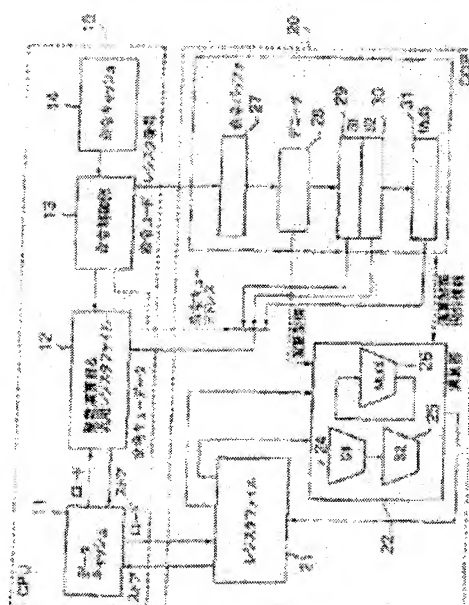
G06F 9/38

(71)Applicant : FUJITSU LTD

(72)Inventor : TSUJI MASAYUKI

本報のハイブラインド印刷機は、色鮮やかで、文字がくっきりと印刷される。また、紙の質感もよく、読者の目にやさしい。本報の印刷機は、色鮮やかで、文字がくっきりと印刷される。また、紙の質感もよく、読者の目にやさしい。

**SOLUTION:** This pipe line processor is provided with first storage means 29 and 30 for storing arithmetic instructions supplied from a central arithmetic unit to arithmetic units, first arithmetic units 24 and 25 for performing arithmetic operations according to the arithmetic instructions stored in the first storage means, a second storage means 31 for storing the arithmetic instruction when the arithmetic operation performed according to the arithmetic instruction requires a long time until the end of the arithmetic operation, and a second arithmetic unit 26 for performing the arithmetic operation according to the arithmetic instruction stored in the second storage means.



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号  
特開2001-243066  
(P2001-243066A)

(43)公開日 平成13年9月7日(2001.9.7)

(51)Int.Cl. <sup>7</sup>	識別記号	F I	テークコード*(参考)
G 0 6 F 9/38	3 1 0	C 0 6 F 9/38	3 1 0 F 5 B 0 1 3
	3 7 0		3 1 0 J
			3 7 0 C

審査請求 未請求 請求項の数6 O L (全 9 頁)

(21)出願番号 特願2000-54832(P2000-54832)

(22)出願日 平成12年2月29日(2000.2.29)

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番  
1号

(72)発明者 辻 雅之

神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内

(74)代理人 100070150

弁理士 伊東 忠彦

Fターム(参考) 5B013 AA12 AA18 DD03

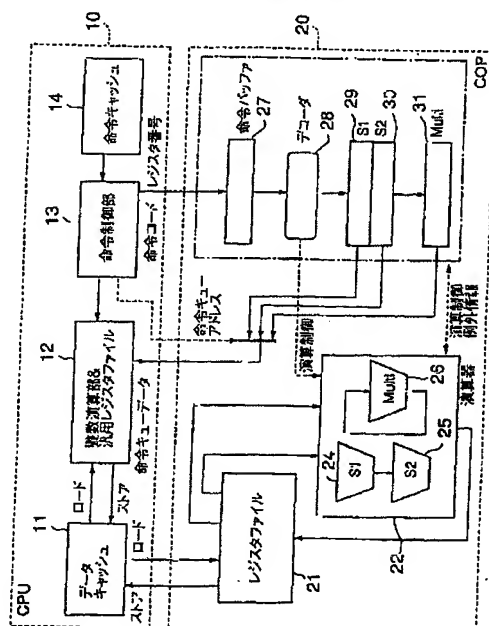
(54)【発明の名称】 パイプライン処理方法並びにその方法を利用するパイプライン処理装置

(57)【要約】

【課題】 命令処理性能の低下を防ぐことができ、消費電力及びコストを削減することが可能なパイプライン処理方法並びにその方法を利用するパイプライン処理装置を提供することを目的とする。

【解決手段】 中央演算装置から演算装置に供給される演算命令を格納する第1格納手段29、30と、第1格納手段に格納された演算命令に従って演算を行なう第1演算器24、25と、演算を行なった演算命令が演算終了までに時間の掛かる演算命令であれば、その演算命令を格納する第2格納手段31と、第2格納手段に格納された演算命令に従って演算終了まで演算を行なう第2演算器26とを有することにより上記課題を解決する。

本発明のパイプライン処理装置の第1実施例のブロック図



【特許請求の範囲】

【請求項1】 中央演算装置と演算装置とを相互接続して演算処理を行なうパイプライン処理方法において、前記演算装置に供給される演算命令を格納する段階と、前記格納された演算命令に従って演算を行い、演算終了までに時間の掛かる演算命令であるか確認する段階と、演算終了までに時間の掛かる演算命令であれば、その演算命令を専用の格納場所にシフトする段階と、前記専用の格納場所にシフトした前記演算命令の演算を演算終了まで行なう段階とを有するパイプライン処理方法。

【請求項2】 演算終了までに時間の掛かる演算命令でなければ、その演算命令の演算結果を順次出力する段階を更に有する請求項1記載のパイプライン処理方法。

【請求項3】 中央演算装置と演算装置とを相互接続して演算処理を行なうパイプライン処理装置において、前記演算装置に供給される演算命令を格納する第1格納手段と、前記第1格納手段に格納された演算命令に従って演算を行なう第1演算器と、前記演算を行なった演算命令が演算終了までに時間の掛かる演算命令であれば、その演算命令を格納する第2格納手段と、前記第2格納手段に格納された演算命令に従って演算終了まで演算を行なう第2演算器とを有するパイプライン処理装置。

【請求項4】 中央演算装置と演算装置とを相互接続して演算処理を行なうパイプライン処理装置において、前記演算装置に供給される演算命令を格納する第1格納手段と、前記第1格納手段に格納された演算命令に従って演算を行なう第1演算器と、前記演算を行なった演算命令が演算終了までに時間の掛かる演算命令であれば、その演算命令を格納する複数の第2格納手段と、前記複数の第2格納手段に格納されている演算命令の発行順序を示す識別手段と、前記識別手段に従って最先に発行された演算命令を選択し、その演算命令に従って演算終了まで演算を行なう複数の第2演算器とを有するパイプライン処理装置。

【請求項5】 中央演算装置と複数の演算装置とを相互接続して演算処理を行なうパイプライン処理装置において、前記複数の演算装置毎に、前記演算装置に供給される演算命令を格納する第1格納手段と、前記第1格納手段に格納された演算命令に従って演算を行なう第1演算器と、前記演算を行なった演算命令が演算終了までに時間の掛かる演算命令であれば、その演算命令を格納する第2格納手段と、

前記第2格納手段に格納されている演算命令の発行順序を示す識別手段と、

前記識別手段に従って最先に発行された演算命令を選択し、その演算命令に従って演算終了まで演算を行なう第2演算器とを有し、

前記複数の演算装置の識別手段に優先順位を設定しておくことを特徴とするパイプライン処理装置。

【請求項6】 前記演算終了までに時間の掛かる演算命令は、演算終了までに複数サイクルの時間を必要とするマルチプル演算命令であることを特徴とする請求項3乃至5何れか一項記載のパイプライン処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、パイプライン処理方法並びにその方法を利用するパイプライン処理装置に係り、特に、中央演算装置(Central Processing Unit)と演算装置とを相互接続して非同期に演算を行なうパイプライン処理方法並びにその方法を利用するパイプライン処理装置に関する。

【0002】

【従来の技術】近年、コンピュータ装置は処理速度の更なる高速化が要求されるようになり、中央演算装置(以下、CPUという。)単体では要求される処理速度を満たさない場合が生じている。そこで、高速演算を実行する為の演算装置を別途設け、CPUと並列に非同期演算を行なわせることによりCPUを補完する処理方法が用いられるようになってきている。このような演算装置としては、例えば浮動小数点演算を行なうコプロセッサ(Co-Processor:以下、COPという。)がある。

【0003】なお、パイプライン処理方法とは、命令の処理過程を複数の段階に分割し、複数の命令をパイプライン的に先回り並行処理する制御方法である。パイプライン処理方法は、各段階の時間間隔毎に命令実行が可能であり、単位時間当りの処理能力が向上する。

【0004】図1は、パイプライン処理装置の一例のブロック図を示す。パイプライン処理装置は、CPU100とCOP200とを含む構成である。CPU100はCOP200と相互接続されており、浮動小数点演算等のCOP200を使用する演算命令があると、その演算命令の命令コードとレジスタ番号とをCOP200に供給する。

【0005】COP200はCPU100から供給される命令コード及びレジスタ番号を命令バッファ230に格納する。命令バッファ230に格納された演算命令は各種パイプラインハザードが解消された段階でパイプラインド演算器220で実行される。演算命令は、パイプラインド演算器220の演算ステージS1、S2等に対応するように命令キュー240、241を伝播していく。

【0006】最終の演算ステージS2では例外チェックが行われ、演算が正常終了しているか判定される。演算が正常終了していれば、演算命令は命令キュー241からデキューされる。また、演算結果はパイプライン演算器220からレジスタファイル210に格納される。演算が正常終了せず例外が検出された場合、演算命令は命令キュー241に残る。そして、例外情報が命令キュー241に登録されると共に、CPU100に対して割り込み要求が行われる。このとき、命令キュー240にある後続の命令は、未完了の命令として命令キュー240に格納される。

【0007】但し、マルチサイクルを必要とするマルチプル演算命令の場合、演算レイテンシが長い為にマルチプル演算命令の情報が、ある命令キュー240、241に複数サイクルの間留まることになる。この間、後続の演算命令は前のステージの命令キュー240又は命令バッファ230で待ちの状態となる。この待ちの状態を最小限にする為、命令バッファ230は待ち命令キュー231、待ち命令キュー232を設けて複数段の構成とし、CPU100が供給する演算命令を格納する。このように、従来のパイプライン処理装置は演算命令と演算の実行との対応を容易とし、例外検出時の割り込み処理も容易としていた。

【0008】

【発明が解決しようとする課題】図2は、パイプライン処理装置の一例のタイムチャートを示す。図2のタイムチャートは、演算命令がマルチプル演算命令a、パイプライン演算命令b、パイプライン演算命令c、パイプライン演算命令d、パイプライン演算命令eの順番で実行される場合について示したものである。

【0009】まず、時刻tにマルチプル演算命令aがCPU100に供給され、命令バッファ230を介して命令キュー240に格納される。しかし、マルチプル演算命令aは実行の終了までにマルチサイクルを必要とするので、時刻t+2から命令キュー240に留まり続ける。

【0010】続いて時刻t+1にはパイプライン演算命令bがCPU100に供給され、命令バッファ230に格納される。時刻t+3では、命令キュー240にマルチプル演算命令aが格納されているのでパイプライン演算命令bが命令バッファ230から待ち命令キュー231に供給される。また、時刻t+4では、パイプライン演算命令bが待ち命令キュー231から待ち命令キュー232に供給された後、待ち命令キュー232に留まり続ける。

【0011】時刻t+2には、パイプライン演算命令cがCPU100に供給され、命令バッファ230に格納される。時刻t+4では、命令キュー240にマルチプル演算命令aが格納されているのでパイプライン演算命令cが命令バッファ230から待ち命令キュー23

1に供給された後、待ち命令キュー231に留まり続ける。

【0012】さらに、時刻t+3にはパイプライン演算命令dがCPU100に供給され、命令バッファ230に格納される。しかし、待ち命令キュー231、232にパイプライン演算命令b、cが留まり続けている為、パイプライン演算命令dは命令バッファ230に留まり続ける。

【0013】従って、時刻t+4にパイプライン演算命令eがCPU100に供給されても命令バッファ230に空きが生じない為、パイプライン演算命令eは処理待ち状態、言い換えればCPUストール状態となる。つまり、従来のパイプライン処理装置は、マルチプル演算命令が供給されると後続の命令がマルチプル演算命令の完了待ちを行なう為、全体の命令処理性能が低下するという問題があった。また、実装されている待ち命令キュー231、232の数を増加すればCPUストール状態を減少させることができるが、消費電力の増加、コストの上昇等の問題があった。

【0014】本発明は、上記の点に鑑みなされたもので、命令処理性能の低下を防ぐことができ、消費電力及びコストを削減することが可能なパイプライン処理方法並びにその方法を利用するパイプライン処理装置を提供することを目的とする。

【0015】

【課題を解決するための手段】そこで、上記課題を解決するため、請求項1記載のパイプライン処理方法は、前記演算装置に供給される演算命令を格納する段階と、前記格納された演算命令に従って演算を行い、演算終了までに時間の掛かる演算命令であるか確認する段階と、演算終了までに時間の掛かる演算命令であれば、その演算命令を専用の格納場所にシフトする段階と、前記専用の格納場所にシフトした前記演算命令の演算を演算終了まで行なう段階とを有することを特徴とする。

【0016】このように、演算終了までに時間の掛かる例えばマルチプル演算命令を実行する場合に、そのマルチプル演算命令を専用の格納場所にシフトする段階を有することにより、後続の演算命令に対する命令処理性能の低下を防ぎ、命令バッファの段数を削減することにより消費電力及びコストの削減が可能である。

【0017】例えば、命令のアウトオブオーダー(Out-Of-Order)完了を許すアーキテクチャでは、必ずしも命令を発行順で実行完了させなくともよい。この場合、本願発明のような制御も可能となる。

【0018】また、請求項2記載のパイプライン処理方法は、演算終了までに時間の掛かる演算命令でなければ、その演算命令の演算結果を順次出力する段階を更に有することを特徴とする。

【0019】このように、演算終了までに時間の掛かる例えばマルチプル演算命令であっても他の演算命令と同

様のタイミングで格納場所を遷移させることができる為、後続の他の演算命令をストールさせることなく演算処理を行なうことが可能である。

【0020】また、請求項3記載のパイプライン処理装置は、前記演算装置に供給される演算命令を格納する第1格納手段（例えば、図3における命令キュー29、30）と、前記第1格納手段に格納された演算命令に従って演算を行なう第1演算器（例えば、図3における演算ステージ24、25）と、前記演算を行なった演算命令が演算終了までに時間の掛かる演算命令であれば、その演算命令を格納する第2格納手段（例えば、図3における命令キュー31）と、前記第2格納手段に格納された演算命令に従って演算終了まで演算を行なう第2演算器（例えば、図3における演算ステージ26）とを有することを特徴とする。

【0021】このように、演算終了までに時間の掛かる例えばマルチプル演算命令を実行する場合に、そのマルチプル演算命令を格納する第2格納手段とそのマルチプル演算命令を行なう第2演算器とを有することにより、後続の演算命令に対する命令処理性能の低下を防ぎ、命令バッファの段数を削減することにより消費電力及びコストの削減が可能である。

【0022】また、請求項4記載のパイプライン処理装置は、前記演算装置に供給される演算命令を格納する第1格納手段と、前記第1格納手段に格納された演算命令に従って演算を行なう第1演算器と、前記演算を行なった演算命令が演算終了までに時間の掛かる演算命令であれば、その演算命令を格納する複数の第2格納手段（例えば、図5における命令キュー37、38）と、前記複数の第2格納手段に格納されている演算命令の発行順序を示す識別手段（例えば、図5におけるアドレス操作ビット39、40）と、前記識別手段に従って最先に発行された演算命令を選択し、その演算命令に従って演算終了まで演算を行なう複数の第2演算器（例えば、図5における演算ステージ35、36）とを有することを特徴とする。

【0023】このように、複数の第2格納手段に格納されている演算命令の発行順序を示す識別手段を有することにより、演算命令の命令発行順に演算終了までに時間の掛かる例えばマルチプル演算命令を実行することが可能となる。

【0024】また、請求項5記載のパイプライン処理装置は、複数の演算装置（例えば、図6におけるCOP50、60）毎に、前記演算装置に供給される演算命令を格納する第1格納手段と、前記第1格納手段に格納された演算命令に従って演算を行なう第1演算器と、前記演算を行なった演算命令が演算終了までに時間の掛かる演算命令であれば、その演算命令を格納する第2格納手段と、前記第2格納手段に格納されている演算命令の発行順序を示す識別手段と、前記識別手段に従って最先に発

行された演算命令を選択し、その演算命令に従って演算終了まで演算を行なう第2演算器とを有し、前記複数の演算装置の識別手段に優先順位を設定しておくことを特徴とする。

【0025】このように、複数の演算装置の識別手段に優先順位を設定しておくことにより、例えば異なる演算装置に同時にマルチプル演算命令が発行した場合に対応することができる。

【0026】また、請求項6記載のパイプライン処理装置は、前記演算終了までに時間の掛かる演算命令は、演算終了までに複数サイクルの時間を必要とするマルチプル演算命令であることを特徴とする。

【0027】このように、マルチプル演算命令を実行する場合に、後続の例えばパイプライン演算命令に対する命令処理性能の低下を防ぐことが可能であり、命令バッファの段数を削減することにより消費電力及びコストの削減が可能である。

【0028】なお、上記括弧内の符号は理解を容易とするために附したものであり、一例にすぎない。

【0029】

【発明の実施の形態】次に、本発明の実施の形態について図面に基づいて説明する。図3は、本発明のパイプライン処理装置の第1実施例のブロック図を示す。パイプライン処理装置は、相互接続されているCPU10とCOP20とを含むように構成される。また、CPU10は、データキャッシュ11、整数演算部&汎用レジスタファイル12、命令制御部13、及び命令キャッシュ14を含むように構成される。COP20はレジスタファイル21、演算器22、命令バッファ27、デコード28、命令キュー29、30、及びマルチプル演算命令用の命令キュー31を含むように構成される。

【0030】CPU10の命令キャッシュ14はプログラムが格納されており、演算命令を命令制御部13に供給する。命令制御部13は演算命令が供給されると、その演算命令が浮動小数点演算等のCOP20を使用する演算命令であるか判断する。COP20を使用する演算命令であると判断すると、その演算命令の命令コードとレジスタ番号とをCOP20の命令バッファ27に供給する。一方、COP20を使用しない整数演算等の演算命令であると判断すると、その演算命令の命令コードとレジスタ番号とを整数演算部&汎用レジスタファイル12に供給する。

【0031】整数演算部&汎用レジスタファイル12はレジスタ番号に従ってデータキャッシュ11からデータを読み出し、命令コードに従ってそのデータの演算を行なう。その後、整数演算部&汎用レジスタファイル12は演算結果をデータキャッシュ11に書込む。

【0032】命令バッファ27は命令制御部13から供給された命令コードとレジスタ番号とを、各種パイプラインハザードが解消された段階でデコード28に供給す

る。即ち、レジスタ干渉、ハードウェアの資源競合がないことを確認する。デコーダ28は供給された命令コードをデコードし、命令キュー29に演算命令を格納すると共に、演算命令及びレジスタ番号を演算器22の演算ステージ24に供給する。なお、各種パイプラインハザードが解消されていない場合、命令バッファ27は命令コード及びレジスタ番号をデコーダ28に供給せず、次サイクルで再度各種パイプラインハザードが解消されているか確認される。

【0033】命令キュー29は、格納されている演算命令をパイプライン形式で演算キュー30に供給する。このとき、演算器22の演算ステージ24に格納されている演算命令及びレジスタ番号は、演算ステージ25に供給される。演算ステージ24から演算命令及びレジスタ番号が供給されると、演算ステージ25は必要なデータをレジスタファイル21から読出し、その演算命令に従って演算を行なう。

【0034】つまり、演算ステージ25に演算命令及びレジスタ番号が供給されると、次サイクルで演算結果が求まる。演算結果が求まると演算ステージ25では演算例外の有無を調べ、演算が正常終了している場合は演算命令がデキューされる。そして、演算器22からレジスタファイル21に演算結果を供給する。一方、演算例外が発生している場合、演算ステージ25及び命令キュー30に演算命令及び演算例外情報を格納して割り込み動作に入る。

【0035】マルチプル演算命令の場合、更に演算が続行するので演算ステージ25及び命令キュー30に格納されている演算命令及び演算例外情報がマルチプル演算命令用の演算ステージ26及び命令キュー31に遷移される。なお、演算の開始時に分かる演算例外、例えばゼロ除算は他のパイプライン演算命令と同様に、例外検出を行なうことも可能である。

【0036】マルチプル演算命令用の演算ステージ26及び命令キュー31に演算終了まで保持される演算命令は、演算終了時に再度演算例外の有無を調べられる。演算例外がなければ、演算ステージ26及び命令キュー31から演算命令がデキューされる。一方、演算例外があれば演算ステージ26及び命令キュー31に演算命令を残し、割り込み動作に入る。なお、演算結果はレジスタファイル21に格納される。

【0037】図3のパイプライン処理装置の処理について図4を参照しつつ説明する。図4は、パイプライン処理装置の一例のタイムチャートを示す。なお、図4では説明に不要な部分を省略している。図4のタイムチャートは、演算命令がマルチプル演算命令a、パイプライン演算命令b、パイプライン演算命令c、パイプライン演算命令d、パイプライン演算命令eの順番で実行される場合について示したものである。

【0038】まず、時刻tにマルチプル演算命令aが命

令キャッシュ14から命令制御部13に供給される。時刻t+1に命令制御部13はマルチプル演算命令aを命令バッファ27に供給する。また、パイプライン演算命令bが命令キャッシュ14から命令制御部13に供給される。

【0039】続いて時刻t+2にマルチプル演算命令aが命令バッファ27から命令キュー29に供給される。命令制御部13はパイプライン演算命令bを命令バッファ27に供給する。また、パイプライン演算命令cが命令キャッシュ14から命令制御部13に供給される。

【0040】時刻t+3では、マルチプル演算命令aが命令キュー29から命令キュー30に供給される。パイプライン演算命令bが命令バッファ27から命令キュー29に供給される。命令制御部13はパイプライン演算命令cを命令バッファ27に供給する。また、パイプライン演算命令dが命令キャッシュ14から命令制御部13に供給される。

【0041】時刻t+4では、マルチプル演算命令aが命令キュー30からマルチプル演算命令用の命令キュー31に供給される。パイプライン演算命令bが命令キュー29から命令キュー30に供給される。パイプライン演算命令cが命令バッファ27から命令キュー29に供給される。命令制御部13はパイプライン演算命令dを命令バッファ27に供給する。また、パイプライン演算命令eが命令キャッシュ14から命令制御部13に供給される。

【0042】時刻t+5では、マルチプル演算命令aが命令キュー31に引き続き格納される。そして、パイプライン演算命令bが演算処理を終了し、デキューされる。パイプライン演算命令cが命令キュー29から命令キュー30に供給される。パイプライン演算命令dが命令バッファ27から命令キュー29に供給される。命令制御部13はパイプライン演算命令eを命令バッファ27に供給する。

【0043】ここで、図2を参照しつつ説明したタイムチャートと比較すると、図2のタイムチャートでは時刻t+5でCPUストールが発生しているが、図4のタイムチャートでは時刻t+5でCPUストールを発生していない。つまり、本発明の第1実施例のパイプライン処理装置では、マルチプル演算命令であっても他のパイプライン演算命令と同様のタイミングで命令キュー29、30を遷移する為、後続のパイプライン命令をストールさせることなく演算処理が可能である。従って、全体としての演算性能が飛躍的に向上すると共に、ストールを極力回避する為に実装されていた複数段の命令バッファの段数を削減することが可能となる。

【0044】なお、後続の演算命令は先行するマルチプル演算命令が実行中であっても演算例外が発生する場合がある。このように後続の演算命令が演算例外を発生し

た場合、先行する演算命令は実行を完了させるか、マルチプル演算命令の検出時或いは後続の演算命令の演算例外の検出時に、命令キューに演算命令、レジスタ番号、及び演算例外情報等を保持することができる。

【0045】図5は、本発明のパイプライン処理装置の第2実施例のブロック図を示す。なお、図5のブロック図はパイプライン演算装置のCOP20を記載したものであり、CPU10の記載を省略している。また、図3のパイプライン演算装置と同様な部分については同一符号を付して説明を省略している。

【0046】図5のパイプライン演算装置は、マルチプル演算命令用の命令キュー37、38及び演算ステージ35、36を設けていることを特徴としている。この場合、マルチプル演算命令用の命令キュー37、38及び演算ステージ35、36に格納されている演算命令の命令発行順序を外部に通知する必要があり、命令キュー37、38にアドレス操作ビット39、40を設けることにより、命令発行順序を明確にすることが可能となっている。

【0047】レイテンシの異なる2つのマルチプル演算命令a、bを異なるマルチプル演算ステージ35、36で実行する場合、この演算ステージ35、36に夫々対応するように命令キュー37、38及びアドレス操作ビット39、40を設ける。アドレス操作ビットとは、命令キューのアドレスを決定する為に使用されるビットである。

【0048】例えば、命令キュー37、38のアドレスに「000」、「001」が割り当てられているものとする。マルチプル演算命令aが発行され、専用の命令キュー37に格納される際に、他方の命令キュー38に付随するアドレス操作ビット40が「0」であればアドレス操作ビット39に「1」をセットする。また、他方の命令キュー38に付随するアドレス操作ビット40が「1」であればアドレス操作ビット39に「0」をセットする。

【0049】そして、アドレス操作ビットが「1」であるマルチプル演算命令が実行完了した場合は、そのアドレス操作ビットを「1」から「0」に変更すると共に、他方の命令キューに付随するアドレス操作ビットを「0」から「1」に変更する。つまり、2つのマルチプル演算命令a、bのうち、先に発行されたマルチプル演算命令用の命令キューに付随するアドレス操作ビットは常に「1」となり、先に発行されたマルチプル演算命令を2ビットのレジスタで知ることが可能である。

【0050】更に、付随するアドレス操作ビットが「1」であるマルチプル演算命令用の命令キューのアドレスは「000」であり、付随するアドレス操作ビットが「0」であるマルチプル演算命令用の命令キューのアドレスは「001」であると決めておくことで、アドレスの昇順に命令キューを読み出せば、常に命令発行順に

マルチプル演算命令の情報を命令キューから読み出すことができる。

【0051】図6は、本発明のパイプライン処理装置の第3実施例のブロック図を示す。なお、図6のブロック図は複数のCOPを有するパイプライン演算装置を記載したものであり、説明に不要な記載を省略している。また、図3のパイプライン演算装置と同様な部分については同一符号を付して説明を省略している。

【0052】図6のパイプライン処理装置は二つのCOP50、60を有し、夫々のCOP50、60にマルチプル演算命令用の命令キュー54、64が含まれる。マルチプル演算命令用の命令キュー54、64は、図6の第2実施例と同様に、命令キュー54、64に付随するようにアドレス操作ビットが設けられている。

【0053】このように、パイプライン演算装置に複数のCOPが含まれる場合、異なるCOPに含まれるマルチプル演算命令用の命令キュー54、64に同時にマルチプル演算命令が格納されることがある。この場合、マルチプル演算命令が命令キュー54、64に同時に発行されたことが保証される限りにおいては、命令キュー54、64に付随するアドレス操作ビットに値をセットする優先順位を決めておく必要がある。この優先順位は、valid生成装置56、66により決定される。なお、その他のタイミングについては、一つのCOP内にマルチプル演算命令用の命令キューが実装されている場合と同様であり説明を省略する。

#### 【0054】

【発明の効果】上述の如く、本発明によれば、マルチプル演算命令を実行する場合に、後続の演算命令に対する命令処理性能の低下を防ぐことが可能であり、命令バッファの段数を削減することにより消費電力及びコストの削減が可能である。

#### 【図面の簡単な説明】

【図1】パイプライン処理装置の一例のブロック図である。

【図2】パイプライン処理装置の一例のタイムチャートである。

【図3】本発明のパイプライン処理装置の第1実施例のブロック図である。

【図4】パイプライン処理装置の一例のタイムチャートである。

【図5】本発明のパイプライン処理装置の第2実施例のブロック図である。

【図6】本発明のパイプライン処理装置の第3実施例のブロック図である。

#### 【符号の説明】

- 10 中央演算装置
- 11 データキャッシュ
- 12 整数演算部&汎用レジスタファイル
- 13 命令制御部

14 命令キャッシュ  
20 コプロセッサ  
21 レジスタファイル  
22 演算器  
27, 51, 61 命令バッファ  
28 デコーダ

【図1】

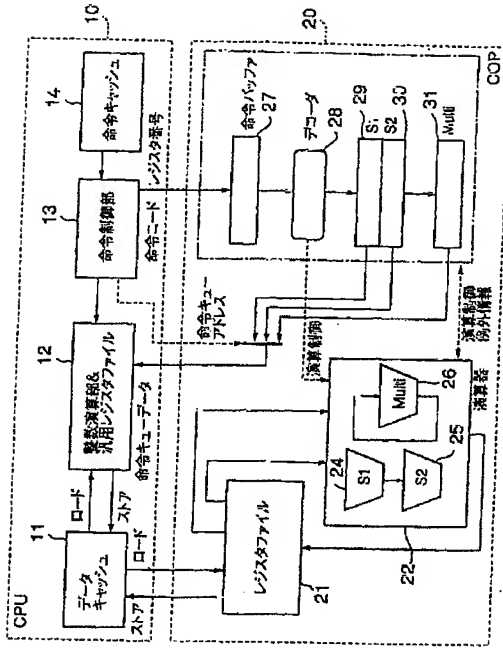
【図2】

[illegible]



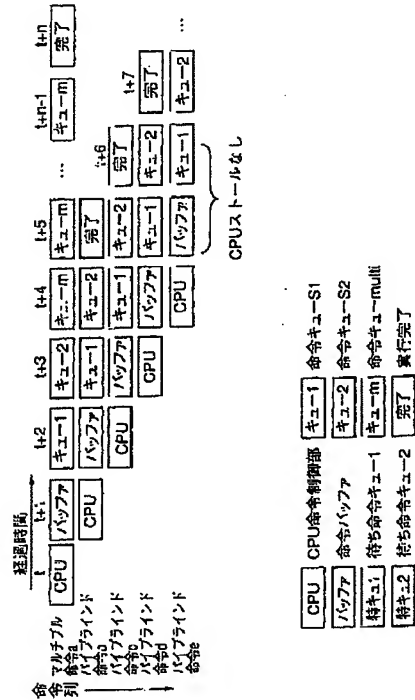
【図3】

本発明のパイプライン処理装置の第1実施例のブロック図



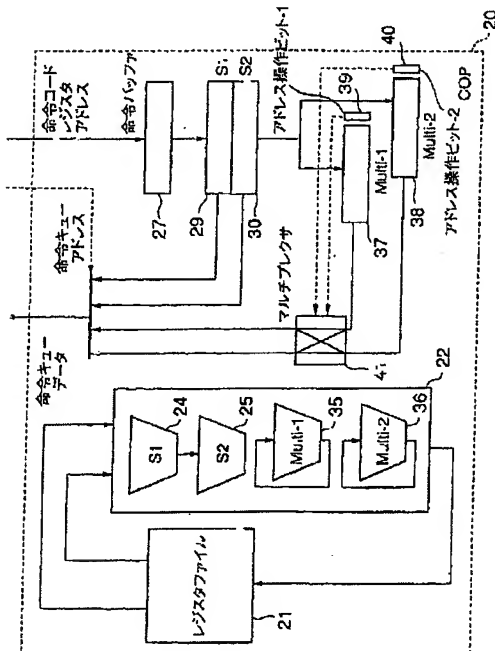
【図4】

パイプライン処理装置の一例のタイムチャート



【図5】

本発明のパイプライン処理装置の第2実施例のブロック図



【図6】

本発明のバイプライン処理装置の第3実施例のブロック図

